

REMARKS

Claim 1 is rejected under 35 U.S.C. 102 (e) as being anticipated by Proteat et al. (U.S. Patent no. 5,970,245; hereinafter Proteat).

The present application uses dynamic linking to facilitate the distribution of DLLs between a host and emulation system to facilitate the operation of a multiprocessor capable In-Circuit emulator.

Applicant's Claim 1 calls for "In a data exchange system for transferring data between a host processor and a target processor comprising:
a data unit on said target processor that transfers said data from said target processor to an emulator and a device driver on said host processor that transfers data from said emulator,
a system for dynamically linking and loading software support for a new target processor comprising:
a target interface module for the host computer that supports the new target processor kind; and
a target interface module for the emulator that supports the new target processor kind."

The examiner attributes more to the Proteat than is taught in the reference. Proteat's patent applies to debugging and tracing shared procedures (DLLs) running on a single computer system. A DLL (Dynamic Link Library) is executable routines available to applications at runtime. A copy of page 111 of The Computer Glossary is attached herewith to aid the examiner's understanding of DLLs. There is no teaching or suggestion of the actual linking process of code to a remote system. Applicant's application is for the process of distributing and linking code dynamically in a remote system. In Proteat's patent the use of the word "target" does not imply remote system, just the DLL under

debug in a shared environment that is limited to a single computer. Applicant's word "target" is specific reference to a remote in-circuit emulation system which provides it's own processing environment independent of the host computer (where our DLL resides but do not run).

The examiner argues that applicants' claims are made by Proteat. This patent describes a method for tracing a DLL's (Dynamic Linked Libraries) API calls within a single processor system. Referring to Fig. 1 it shows a single computer with a single processor 12. The description of Fig. 1 is that it is a block diagram that illustrates an exemplary hardware environment of the present invention. The memory includes an operating system, software applications and a trace DLL and a target DLL. There is only one processor. The examiner references col. 2, lines 36-47 but nowhere does it teach or discuss more than one processor. There is no teaching of a data exchange system for transferring data between a host processor and a target processor. There is no a data unit on said target processor that transfers said data from said target processor to an emulator and a device driver on said host processor that transfers data from said emulator. The examiner also references Col. 3, lines 13-23 but there is no teaching of more than one processor or of a data exchange system for transferring data between a host processor and a target processor. There is a target procedure in a target DLL but no target processor or loading emulation software support for a new target processor. Col. 2, lines 48-64 do not mention multiprocessor including a new target processor. The reference patent describes a method for tracing a DLL's (Dynamic Linked Libraries) API calls within a single processor system. In view of the above applicant's claim 1 is deemed allowable.

Claim 2 calls for “ A method for at time of use linking and loading of emulation software for one or more debuggers on a host computer to communicate with a mix of target processors via a JTAG debug link and emulator comprising the steps of: connecting a debugger for each processor to a target interface for that kind of processor; determining if there is support for that kind of processor in the emulator by the target interface communicating with a dynamic loader on the host computer; if not support loading a target interface into the emulator and connecting to an already running emulation software on the host computer; and connecting the target interface software on the emulator to the target interface software on the host computer.”

The specification of the reference patent of Proteat has been reviewed and this patent describes a method for tracing a DLL's (Dynamic Linked Libraries) API calls within a single processor system. The patent does not teach linking and loading of emulation software for one or more debuggers on a host computer to communicate with a mix of target processors via a JTAG debug link and emulator. There is no step in the reference of connecting a debugger for each processor to a target interface for that kind of processor. There is no determining if there is support for that kind of processor in the emulator by the target interface communicating with a dynamic loader on the host computer and no step of if not support loading a target interface into the emulator and connecting to an already running emulation software on the host computer; and connecting the target interface software on the emulator to the target interface software on the host computer. The examiner references Col. 4, lines 21-63 but that text only describes a method for tracing DLL API calls within a single processor. JTAG interconnect is known in and described in IEEE standard 1149.1.

Claim 3 calls for the method of Claim 2 wherein said steps are repeated for each debugger, for each kind of processor on the target system. Claim 3 is therefore deemed allowable for at least the same reasons as Claim 2. It also calls for repeating the steps for each kind of processor. There is only a single processor in the Proteat reference.

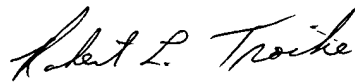
Claim 4 calls for “The method of Claim 2 wherein said determining step includes communicating using ECOM modules on the host computer and emulator over a host computer to emulator connection.” Claim 4 dependent on Claim 2 is deemed allowable for at least the same reasons as Claim 2. The claim further describes using

ECOM modules on the host computer and emulator in a host computer to emulator connection. No such connection is suggested in the Proteat reference. ECOM is well known term of art that refers to common embedded component module as shown and described in Fig. 1.

Claim 5-8 are deemed allowable over Proteat for the reasons presented above.

In view of the above applicants' claims 1-8 are deemed allowable and an early notice of allowance of these claims is deemed in order and is respectfully requested.

Respectfully submitted;

A handwritten signature in cursive script, reading "Robert L. Troike".

Robert L. Troike (Reg. 24183)

Tel. No. 301-259-2089

The Computer Glossary

THE COMPLETE ILLUSTRATED DICTIONARY

**Seventh Edition
by Alan Freedman**

More than 6,000 meaningful terms! Covers computer jargon and concepts, from personal computers to mainframe. Over 3,000 new and revised entries. Plus over 200 photos and illustrations.

For beginners, power users, and systems people. For EVERYONE!

Information on DOS, Windows, Mac and UNIX, plus networking, client/server, graphics, multimedia, object-oriented programming, standards, major products, vendors, industry profiles — and much more!

Cross-referencing and phonetic entries give you faster access to terms and technology than ever before.

**"The Rolls-Royce of computer dictionaries."
— *Toronto Computes***

praise from
PC
Magazine

on the Sixth Edition
September 28, 1993

"Its score
of 90% was
the highest
for the
dictionaries
we tested."

distributed database A database physically stored in two or more computer systems. Although geographically dispersed, a distributed database system manages and controls the entire database as a single collection of data. If redundant data is stored in separate databases due to performance requirements, updates to one set of data will automatically update the additional sets in a timely manner.

distributed file system Software that keeps track of files stored across multiple networks. It converts file names into physical locations.

distributed function The distribution of processing functions throughout the organization.

distributed intelligence The placing processing capability in terminals and other peripheral devices. Intelligent terminals handle screen layouts, data entry validation and other pre-processing steps. Intelligence placed into disk drives and other peripherals relieves the central computer from routine tasks.

distributed logic See *distributed intelligence*.

distributed processing Also called *distributed computing*, it is a system of computers connected by a communications network. The term is used loosely to refer to any computers with communications between them. However, in true distributed processing, each computer system is sized to handle its local workload, and the network has been designed to support the system as a whole. Contrast with *centralized processing* and *decentralized processing*.

dithering In computer graphics, the creation of additional colors and shades from an existing palette. In monochrome displays, shades of grays are created by varying the density and patterns of the dots. In color displays, colors and patterns are created by mixing and varying the dots of existing colors.

Dithering is used to create a wide variety of patterns for use as backgrounds, fills and shading, as well as for creating halftones for printing. It is also used in anti-aliasing.

divide overflow A program error in which a number is accidentally divided by zero or by a number that creates a result too large for the computer to handle.

DL/1 (Data Language 1) The database language in IMS.

DLC (1) (Data Link Control) See *data link* and *OSI*.

(2) (Data Link Control) The protocol used in IBM's Token Ring networks.

(3) (Digital Loop Carrier) See *loop carrier*.

DLC chip Any one of several Intel-compatible CPUs from Cyrix Corporation. See *486DLC*.

DLL (Dynamic Link Library) Executable routines available to applications at runtime. They are typically written in reentrant code so they can serve more than one application at the same time.

Under DOS, TSRs have been used as a way of adding functionality at runtime. They remain in memory, intercept upon certain conditions, then perform their function. TSRs have never been formally sanctioned and are prone to conflict. Windows, however, has adopted the dynamic link library, or DLL, method as a standard way of creating new functionality that can be shared in the system.

DMA (Direct Memory Access) Specialized circuitry or a dedicated microprocessor that transfers data from memory to memory without using the CPU. On PCs, there are eight DMA channels. Most sound cards are set to use DMA channel 1.

DME (Distributed Management Environment) A set of programs from OSF that provides coherent management of networks and systems.

DMI (Desktop Management Interface) A management system for PCs that provides a bi-directional path to interrogate all the hardware and software components within a

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.